netidee

PROJEKTE

Impressum Extractor

Entwicklerdokumentation | Call 19 | Projekt ID 7285

1 Projektziel

2012 wurde das Projekt Watchlist Internet (WL) durch eine netidee-Förderung ins Leben gerufen. Ziel war und ist es, Privatpersonen und Unternehmen über aktuelle Online-Betrugsmaschen zu informieren und ihnen mit wertvollen Tipps dabei zu helfen, sich vor Internetfallen zu schützen. Mittlerweile ist die Watchlist Internet die größte Präventionsplattform gegen Internetbetrug im deutschsprachigen Raum. 2024 wurden rund 4 Mio. Website-Besucher:innen sowie mehr als 1 Mio. App-Aufrufe verzeichnet, gleichzeitig wurden 183 Warnartikel geschrieben und mehr als 7.000 betrügerische Domains auf die Warnlisten der Watchlist Internet gesetzt.

Um diesen Output zu generieren, sichtet die Redaktion täglich hunderte User-Meldungen, prüft aber auch tausende maschinell detektierte Websites, KI-basierte Risikobewertungen des Fake-Shop Detector (FSD) und Suchergebnisse von unterschiedlichen Crawlern. Mit dieser steigendenden Datenmenge stößt das Team der Watchlist Internet an ihre Grenzen. Gleichzeitig muss das hohe Vertrauen, das die WL sowohl in der Öffentlichkeit als auch bei Partner:innen genießt, gewahrt werden.

Die Anschlussförderung des Projektes „Watchlist Internet“ setzte genau hier an. Ziel des Projektes war es, Lösungen zu entwickeln, die das Team der Watchlist Internet dabei unterstützen, den Qualitätsanspruch bei steigender Quantität zu sichern. Entwickelt wurden teil-automatisierte und regelbasierte Workflows, um die Arbeitsprozesse der Watchlist Internet in Kombination mit Expertenwissen nach Human-in-the-Loop (HITL) Prinzipien zu optimieren. Gleichzeitig wurden insbesondere der HITL-Ansatz sowie die verwendeten Methoden der Entscheidungsfindung hinsichtlich ethischer Standards evaluiert.

Viele Websites stellen Pflichtangaben (Impressum/Kontakt/Offenlegung) auf unterschiedlichen Unterseiten und in sehr unterschiedlichen Formaten bereit. Entwickelt wurde dafür ein Tool (ImpressumExtractor), das Impressumsdaten automatisiert auffindet und strukturiert extrahieren, um eine Weiterverarbeitung (z. B. Monitoring, Datenanalyse, Dokumentation) zu ermöglichen.

2 Impressum Extractor

https://github.com/oiat/impressum-extractor

Der Impressum Extractor ist ein auf Python basierendes Tool zum automatisierten Auffinden und Extrahieren von Impressumsdaten auf Webseiten mithilfe eines Web-Crawlers und eines LLMs (Google Gemini). Ziel ist es, den Prozess der Website-Überprüfung zu beschleunigen und die Impressums-Daten automatisiert darin einfließen lassen zu können.

Projektübersicht

Der Impressum Extractor automatisiert konkret folgende Schritte:

1. Nimmt eine **Start-URL** entgegen.
2. **Crawlt** die Website und sucht nach Impressum-/Kontakt-Seiten.
3. Lädt HTML-Seiten, **bereinigt** den Inhalt und sammelt potenziell relevante Links.
4. Übergibt den bereinigten Text an das **LLM (Gemini)**.
5. Erzwingt über Prompt und JSON-Schema eine Ausgabe als **valides JSON**.
6. Speichert das Ergebnis in der konfigurierten Ausgabedatei (Standard: `result.json`).

Voraussetzungen

Damit der Impressum Extractor funktioniert sind folgende Voraussetzungen notwendig:

- Python 3.9 oder höher
- API-Zugangsdaten (API-Key) für Gemini (Google Gemini / Google AI Studio)
- Abhängigkeiten in `requirements.txt`

Installation

1. Repository klonen:

```
git clone https://github.com/oiat/impressum-extractor.git
cd ImpressumExtractor
```

2. Virtuelle Pyhton-Umgebung erstellen und aktivieren

```
python -m venv .venv
source .venv/bin/activate   # macOS/Linux
venv\Scripts\activate       # Windows
```

3. Abhängigkeiten installieren

```
pip install -r requirements.txt
```

Konfiguration

Standard-Konfiguration: `settings.toml`
Kopiere example.settings.toml nach `settings.toml` und fülle `gemini.api_key` mit deinem API-Key.
Wichtige Einstellungen:

- `gemini.model` — zu verwendendes LLM-Modell
- `gemini.api_key` — API-Key für das LLM
- `impressum.prompt_path` — Pfad zur Prompt-Vorlage (Standard: `prompts/prompt.txt`)
- `impressum.scheme_path` — Pfad zum JSON-Schema (Standard: `schemas/scheme.json`)
- `impressum.output_path` — Ausgabe-Datei (Standard: `result.json`)

Nutzung

Einfache Ausführung für eine URL: `python main.py --url` https://example.com

Das Ergebnis wird in der Datei gespeichert, die in `settings.toml` unter `impressum.output_path` konfiguriert ist (Standard: `result.json`).

Projektstruktur

```
ImpressumExtractor/
├─── src/
│    ├─── main.py                          # Einstiegspunkt des Programms
│    ├─── services/impressum_extractor.py # Suche, Vorverarbeitung und LLM-Auswertung.
│    ├─── services/website_crawler.py     # HTTP-Requests und HTML-Bereinigung.
│    ├─── prompts/prompt.txt              # Prompt-Template für das LLM.
│    ├─── schemas/scheme.json             # Erwartetes JSON-Schema für die LLM-Ausgabe.
│    ├─── settings.toml / example.settings.toml         # Konfiguration.
├─── example.settings.toml                # Template für Umgebungsvariablen
├─── requirements.txt                     # benötigte Python-Pakete.
└─── README.md                            # Projektbeschreibung
```

Tipps für Entwickler

- Wenn sich das gewünschte Ausgabeformat ändert, müssen Prompt `prompts/prompt.txt` und Schema `schemas/scheme.json` angepasst werden.
- Das Tool setzt auf Schema-konformes JSON. Bei Änderungen am Schema sollten Tests/Beispiel-URLs erneut geprüft werden.
- Für Debugging in `settings.toml` das log-level auf DEBUG setzen.
- Gemini-Aufrufe können Kosten verursachen und Rate-Limits haben.

Beispiel Ausgaben bei Standardnutzung

- https://agi-akku.de

{
 "company_name": "AGI Angela Gutzeit Industrievertretungen GmbH",
 "address": "Curslacker Deich 194a 21039 Hamburg Deutschland",
 "email": "verkauf@agi-akku.de",
 "phone": "04152 / 887690",
 "vat_id": "DE 118634419",
 "found": true,
 "notes": "Alle relevanten Impressumsdaten wurden erfolgreich aus dem Text extrahiert."
}

- https://shop.orf.at

{
 "company_name": "ORF Marketing & Creation GmbH & Co KG",
 "address": "1136 Wien, Hugo-Portisch-Gasse 1",
 "email": "shop@orf.at",
 "phone": null,
 "vat_id": "ATU 66572924",
 "found": true,
 "notes": "Die Daten wurden erfolgreich aus dem Abschnitt 'Offenlegung und Informationen gem. § 25 MedienG, § 5 ECG und § 14 UGB' extrahiert. Eine Telefonnummer war im Text nicht explizit angegeben, lediglich E-Mail-Adressen für den Support."
}

- https://keycity.com/

{
 "company_name": "E&R Software GmbH",
 "address": "Franz-Baumann-Weg 22/18, 6020 Innsbruck, Österreich",
 "email": "office@er-games.at",
 "phone": "+43 720 519517",
 "vat_id": "ATU72245368",
 "found": true,
 "notes": "Die Daten wurden aus dem Impressums-Abschnitt extrahiert. Es wird zusätzlich eine Support-E-Mail (office@keycity.com) und eine leicht abweichende Adresse im Footer (ohne Top-Nummer) erwähnt."
}

- https://mooris.de/

{
 "company_name": "Mooris.de",
 "address": null,
 "email": "hallo@mooris.de",
 "phone": "+49 30 22011808",
 "vat_id": null,
 "found": true,
 "notes": "Ein Firmenname (Mooris.de), eine Telefonnummer und eine E-Mail-Adresse wurden im Text gefunden. Eine vollständige physische Adresse sowie eine Umsatzsteuer-Identifikationsnummer fehlen in diesem Website-Auszug. Der Text deutet darauf hin, dass das Unternehmen ursprünglich aus der Schweiz stammt, aber eine deutsche Support-Nummer nutzt."
}

Relevante API-Dokumentationen

Gemini API: https://ai.google.dev/gemini-api/docs?hl=de
Gemini API Key: https://aistudio.google.com/api-keys

Lizenz